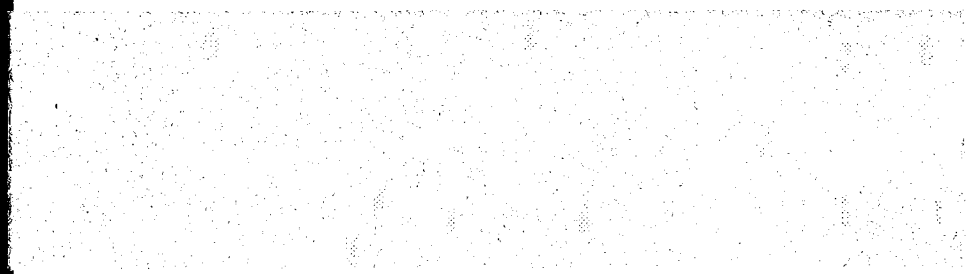

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ARC 275 (Rev Feb 81)

Vector Performance Analysis of Three Supercomputers: Cray-2, Cray Y-MP, and ETA10-Q

R. A. Fatoohi [1]

Report RNR-89-003, April 1989

Sterling Software
NASA Ames Research Center
Moffett Field, California 94035

Abstract. This paper presents the results of a series of experiments to study the single processor performance of three supercomputers: Cray-2, Cray Y-MP, and ETA10-Q. The main object of this study is to determine the impact of certain architectural features on the performance of modern supercomputers. Features such as clock period, memory links, memory organization, multiple functional units, and chaining are considered here. A simple performance model is used to examine the impact of these features on the performance of a set of basic operations. The results of implementing this set on these machines for three vector lengths and three memory strides are presented and compared. For unit stride operations, the Cray Y-MP outperformed the Cray-2 by as much as three times and the ETA10-Q by as much as four times for these operations. Moreover, unlike the Cray-2 and ETA10-Q, even-numbered strides do not cause a major performance degradation on the Cray Y-MP. Two numerical algorithms are also used for comparison. For three problem sizes of both algorithms, the Cray Y-MP outperformed the Cray-2 by over two times and the ETA10-Q by four to eight times.

---

[1] This research was supported by NASA Contract No. NAS2-11555 while the author was an employee of Sterling Software under contract to the NAS Applied Research Office at NASA Ames Research Center.

1. Introduction

The current generation of supercomputers have fast clocks, several powerful processors, and large memories. These machines are faster than the ones developed 15 years ago simply because they have faster clock rates, by three times comparing the Cray-2 with the Cray-1, and many processors, up to eight on the Cray Y-MP. However, the computational engines of these machines have not changed that much for the last 15 years; see [7] for details. Each CPU of the current machines has an internal structure very similar to the old ones. The Cray-1, introduced in 1976, has evolved into the Cray-2 and Cray Y-MP, and the STAR-100, introduced in 1974, has evolved into the ETA10 machines. Hardware and software means to promote parallelism on vector machines were introduced long ago, and are available in different forms on current machines.

There has been a lot of interest in performance comparison of different supercomputers. Among the recent studies is the work by Lubeck et al. [9] to compare the performance of the Japanese Fujitsu VP-200 and Hitachi S810/20 supercomputers with the Cray X-MP/2 using the Los Alamos benchmark set. Anderson et al. [1] compared the performance of the Cray-2 with the Cray X-MP/2 using a set of 24 computational kernels, large linear equation solvers, and 2-D FFT routines. Kampe and Nguyen [8] compared the performance of the Cray-2 with the Cray X-MP/4 for a class of seismic data processing algorithms. These experiments showed comparative results for these machines.

The purpose of this paper is two-fold: to study the impact of certain architectural features on the performance of current supercomputers and to compare the performance of three machines using a set of basic operations and two numerical

algorithms. The three machines of interest are all high-performance vector processors that employ pipeline techniques in both scalar and vector operations. All these machines are currently operational at the Numerical Aerodynamic Simulation (NAS) System Division at NASA Ames Research Center.

This paper presents the results of a series of experiments to study the single processor performance of three supercomputers: Cray-2, Cray Y-MP, and ETA10-Q. The machine architectures are briefly described in section 2. The results of implementing a set of basic operations on these machines for three vector lengths and three memory strides are presented in section 3. Section 4 presents a simple performance model to determine the impact of certain architectural features on the performance of these machines. Section 5 presents a comparison of performance of these machines. The results of implementing two numerical algorithms on the three machines are presented and compared in section 6. Finally section 7 contains some concluding remarks.

2. The architectures

2.1. The Cray-2

The Cray-2 is an MIMD supercomputer with four CPUs, a foreground processor which controls I/O, and a main memory [2]. The main memory has 256 Mwords (64-bit) organized in four quadrants of 32 banks each. Each CPU has access to one quadrant during each clock period. The clock period (CP) is 4.1 nanoseconds. The results reported here were obtained using the Cray-2 with a faster main memory (80 ns DRAM) at NASA Ames Research Center.

2.2. The Cray Y-MP

The Cray Y-MP architecture is an evolutionary step from the Cray X-MP series of supercomputers [3]. It has eight CPUs, 32 Mwords (64-bit) of main memory, and 256 Mwords of SSD memory. The main memory is organized in four sections of 64 banks each. The first Cray Y-MP was delivered to NASA Ames Research Center in August 1988 with a 6.3 nanosecond clock period.

Both the Cray-2 and Cray Y-MP run the UNICOS operating system which is based on UNIX System V. Currently there are two Fortran vectorizing compilers on both machines, the CFT2 and CFT77 compilers. This work was performed using the CFT77 compiler on UNICOS Release 4.0.

2.3. The ETA10-Q

The ETA10-Q is an air-cooled model of the ETA10 supercomputer with a 19 nanosecond clock period [4]. The machine that was delivered to NASA Ames Research Center in April 1988 has one CPU and 64 Mwords (64-bit) of shared memory. The CPU has an internal structure very similar to a two-pipe Cyber 205

with 4 Mwords of processor memory (local memory). This is the first machine to be delivered with the UNIX operating system. The main programming language is Fortran 77 with the VAST-2 vectorizer. This work was performed using this machine during the acceptance period.

Some of the main features of these machines are listed in Table 1.

Table 1. Machine characteristics (single CPU).

Feature	Cray-2	Cray Y-MP	ETA10-Q
Clock Period (ns)	4.1	6.3	19
Peak Performance (Mflops)	487.8	317.5	210.5
No. of Add and multiply units	2	2	2 × 2
Vector registers	8 × 64	8 × 64	None
Local memory	16K (ECL)	None	4M (SRAM)
Central memory	256M (DRAM)	32M (ECL)	64M (DRAM)
Memory paths	1 (64-bit)	3 (64-bit)	3 (128-bit)
Memory banks	128	256	128 (local)
Chaining	No	Yes	Linked triadic ops.
Memory latency (CPs)	45	5	23 (local)
Operating system	UNICOS	UNICOS	UNIX System V
Compiler	CFT77	CFT77	FTN77/VAST-2

3. Test problems: basic operations

A set of basic vector operations was developed specifically for this study. A list of these operations is given in Table 2, where a_i, b_i, c_i, d_i, and e_i are vectors and α, β, γ, and δ are scalars. These basic operations were chosen so as to encompass a different number of floating point and vector memory access operations using a few scalar and vector operands; at most four scalars and four vectors are used in each basic operation. Not all possible combinations of floating point and memory operations are considered here. Also, only add and multiply operations are considered as operators, since other operations are implemented differently on different machines. A straightforward double loop was used to perform each basic operation, as follows:

$J = 1, 100000$

$I = 1, S \times N, S$

where N is the vector length and S is the stride. Vector lengths of 64, 128, and 256 and strides of 1, 2, and 4 are considered here. These vector lengths match the length of the vector registers on Cray machines. The strides are probably the most used ones in real application codes.

Table 2. List of basic operations.

Loop No.	Operation
1	$a_i = \alpha\ b_i$
2	$a_i = b_i\ c_i$
3	$a_i = \alpha\ (b_i + c_i)$
4	$a_i = b_i\ (c_i + d_i)$
5	$a_i = \alpha\ b_i + \beta\ c_i$
6	$a_i = \alpha\ b_i + c_i\ d_i$
7	$a_i = b_i\ c_i + d_i\ e_i$
8	$a_i = \alpha\ b_i + \beta\ c_i + \gamma\ d_i$
9	$a_i = \alpha\ b_i + \beta\ c_i + \gamma\ d_i + \delta\ e_i$

Tables 3 through 5 contain the processing rates (in Mflops) and the execution times (in CPs) of the nine loops on the Cray-2, Cray Y-MP, and ETA10-Q, respectively. Also, the average values for the nine loops are given in these tables for comparison. The results listed in these tables were determined as follows: the total execution time of the outer loop for each basic operation was measured, and the result was divided by the number of the outer loop iterations to yield an average time for the inner loop. The processing rate was computed by taking the ratio of the number of floating point operations in each outer loop to the average time for that loop. The average time for each inner loop was divided by the clock period for each machine to compute the number of clock periods. The entire test was repeated many times, and the best case times are listed in Tables 3 to 5. These best case times can be considered as near optimal values for these operations. The fluctuation of the timing results was mostly significant on the Cray-2, where up to 50% difference was noted depending on system load. The best case times were chosen because they include minimum software overhead, which may help in the performance analysis.

Table 8. Performance of basic operations on one processor of the Cray-2.

Loop	Stride 1		Stride 2		Stride 4	
No.	Proc. rate (Mflops)	Exec. time (CPs)	Proc. rate (Mflops)	Exec. time (CPs)	Proc. rate (Mflops)	Exec. time (CPs)
			$I = 1, S \times 64, S$			
1	52.1	300	25.1	622	16.0	976
2	38.3	407	18.1	863	11.2	1398
3	65.0	480	33.2	939	21.1	1476
4	63.6	490	28.1	1110	17.2	1820
5	100.7	466	49.8	939	31.7	1476
6	84.1	556	39.5	1188	24.6	1902
7	76.5	612	32.7	1432	20.1	2327
8	133.4	585	65.5	1193	41.0	1905
9	162.6	673	76.1	1437	46.9	2332
Avg.	86.3	507	40.9	1080	25.5	1735
			$I = 1, S \times 128, S$			
1	54.7	571	25.1	1244	15.8	1976
2	43.2	724	18.0	1729	11.2	2795
3	70.3	880	33.2	1878	21.1	2963
4	69.2	902	28.1	2222	17.1	3646
5	106.1	883	49.8	1878	31.7	2956
6	89.6	1046	39.5	2373	24.6	3805
7	79.7	1176	32.7	2863	20.1	4661
8	139.6	1117	65.4	2385	40.7	3832
9	167.0	1307	76.0	2876	46.7	4683
Avg.	91.1	956	40.9	2161	25.4	3480
			$I = 1, S \times 256, S$			
1	55.1	1132	24.9	2502	16.0	3898
2	43.4	1729	18.1	3454	11.1	5615
3	70.9	1763	32.5	3839	21.2	5900
4	70.5	1771	28.2	4434	17.1	7322
5	106.3	1763	49.9	3754	31.7	5900
6	91.1	2056	39.2	4773	24.6	7602
7	86.8	2159	34.7	5395	21.1	8878
8	163.9	1905	71.9	4341	43.4	7185
9	191.0	2288	80.7	5417	48.6	8990
Avg.	97.7	1841	42.2	4212	26.1	6810

Table 4. Performance of basic operations on one processor of the Cray Y-MP.

Loop No.	Stride 1		Stride 2		Stride 4	
	Proc. rate (Mflops)	Exec. time (CPs)	Proc. rate (Mflops)	Exec. time (CPs)	Proc. rate (Mflops)	Exec. time (CPs)
			$I = 1, S \times 64, S$			
1	114.4	89	110.9	92	78.4	130
2	102.5	98	90.6	113	52.2	195
3	212.1	95	186.0	110	104.3	195
4	144.5	141	138.2	148	78.5	259
5	207.8	146	207.8	146	154.9	197
6	207.8	146	207.1	148	117.6	259
7	207.8	146	167.4	183	93.8	325
8	247.8	205	247.0	206	193.8	262
9	242.0	294	243.0	292	216.5	329
Avg.	187.4	151	177.6	160	121.1	239
			$I = 1, S \times 128, S$			
1	114.6	178	111.0	183	78.4	259
2	102.8	197	90.4	225	52.2	389
3	212.4	192	184.4	221	104.4	389
4	144.6	281	138.9	292	78.4	519
5	207.8	294	207.6	294	155.0	394
6	207.8	294	206.7	295	117.5	519
7	207.6	294	168.0	363	94.1	648
8	246.5	413	240.8	422	193.5	525
9	242.3	587	243.3	584	216.6	657
Avg.	187.4	303	176.8	320	121.1	478
			$I = 1, S \times 256, S$			
1	136.3	298	130.2	313	78.2	519
2	112.7	360	93.3	435	52.2	778
3	243.0	335	185.1	440	104.2	779
4	171.2	475	143.1	568	78.1	1040
5	207.8	587	207.7	587	155.6	783
6	207.8	587	206.9	589	117.2	1040
7	207.7	587	167.7	727	94.0	1297
8	245.0	829	244.1	832	193.7	1049
9	242.3	1175	239.7	1187	216.2	1316
Avg.	197.1	581	179.7	631	121.1	956

Table 5. Performance of basic operations on the ETA10-Q.

Loop No.	Stride 1		Stride 2		Stride 4	
	Proc. rate (Mflops)	Exec. time (CPs)	Proc. rate (Mflops)	Exec. time (CPs)	Proc. rate (Mflops)	Exec. time (CPs)
			$I = 1, S \times 64, S$			
1	63.1	53	9.1	368	7.0	481
2	52.6	64	9.1	368	5.0	673
3	74.8	90	17.5	384	9.6	704
4	46.8	144	15.0	448	7.2	931
5	58.8	172	22.3	452	13.7	736
6	53.9	187	21.8	464	10.7	943
7	48.6	208	17.5	576	8.6	1172
8	58.4	288	25.9	650	15.5	1089
9	70.1	336	29.5	801	17.1	1377
Avg.	58.6	171	18.7	501	10.5	901
			$I = 1, S \times 128, S$			
1	84.2	80	11.7	576	8.4	801
2	84.0	80	11.7	576	6.1	1105
3	140.2	96	23.4	576	11.9	1137
4	76.5	176	19.1	704	8.9	1521
5	97.1	208	28.1	720	16.6	1217
6	91.8	220	27.4	736	12.4	1633
7	75.7	267	23.4	865	10.3	1953
8	87.6	384	32.4	1041	19.1	1761
9	101.1	467	38.8	1216	21.7	2177
Avg.	93.1	220	24.0	779	12.8	1478
			$I = 1, S \times 256, S$			
1	90.9	148	13.8	976	9.1	1473
2	93.5	144	13.8	976	6.9	1943
3	171.3	157	27.2	993	13.4	2017
4	93.5	288	21.6	1248	10.2	2636
5	131.8	307	32.4	1248	18.8	2145
6	135.2	299	32.4	1249	14.5	2786
7	88.1	459	26.3	1537	12.0	3367
8	107.0	630	36.9	1825	22.4	3011
9	122.7	768	44.4	2124	25.5	3693
Avg.	114.9	356	27.6	1353	14.8	2563

4. Performance model

There are certain architectural features, beside pipelining, that promote parallelism on vector machines. Among these are multiplicity of functional units, overlapping of CPU and memory operations, multiplicity of memory ports, and chaining. In order to determine the impact of these features on the performance of these machines, a simple performance model was developed. This model is based on estimating the number of *effective* floating point and memory access operations. The effective operations are the ones which dominate the execution time. The operations that are overlapped with, or hidden behind, the effective operations are considered ineffective. For example, if a floating point operation can be executed concurrently with a memory access operation and the latter takes more time to finish than the former, then the memory operation is considered as an effective operation while the floating point operation is ineffective.

The processing rate of a code can be modeled by

$$f_p = \frac{N_f}{N_{eo} CP},$$

where f_p is the estimated processing rate, N_f is the number of floating point operations in the code, N_{eo} is the total number of effective floating point and memory access operations in that code, and CP is the clock period. Tables 6 through 8 contain the number of actual operations, number of effective operations, estimated processing rate, estimated processing rate for specific values, and measured processing rate for the basic operations on one processor of the Cray-2, Cray Y-MP, and ETA10-Q, respectively. The results for vectors of length 256 were considered for comparison because this vector is long enough to give near optimal performance on

the three machines (about 90% of maximum performance). The performance does not significantly increase on the Cray machines once vector lengths exceed 64, provided that they are multiples of 64.

Table 6. The number of actual operations, number of effective operations, estimated processing rate, estimated processing rate for $m = 1.5f$ (f is a floating point operation and m is a memory access operation), and measured processing rate for the basic operations on one processor of the Cray-2 for vector length 256 and stride 1.

Loop No.	Actual operations	Effective operations	Estimated proc. rate (f_p)	f_p for $m=1.5f$	Measured proc. rate
1	$f + 2m$	$f + 2m$	$f / (f + 2m)CP$	61.0	55.1
2	$f + 3m$	$f + 3m$	$f / (f + 3m)CP$	44.3	43.4
3	$2f + 3m$	$2f + 3m$	$2f / (2f + 3m)CP$	75.0	70.9
4	$2f + 4m$	$f + 4m$	$2f / (f + 4m)CP$	69.7	70.5
5	$3f + 3m$	$2f + 3m$	$3f / (2f + 3m)CP$	112.6	106.3
6	$3f + 4m$	$2f + 4m$	$3f / (2f + 4m)CP$	91.5	91.1
7	$3f + 5m$	$2f + 5m$	$3f / (2f + 5m)CP$	77.0	86.8
8	$5f + 4m$	$2f + 4m$	$5f / (2f + 4m)CP$	152.4	163.9
9	$7f + 5m$	$2f + 5m$	$7f / (2f + 5m)CP$	179.7	191.0

Table 7. The number of actual operations, number of effective operations, estimated processing rate, estimated processing rate for $m = f$, $c = 0.1f$ (f, m, and c are floating point, memory access, and chaining operations), and measured processing rate for the basic operations on one processor of the Cray Y-MP for vector length 256 and stride 1.

Loop No.	Actual operations	Effective operations	Estimated processing rate (f_p)	f_p for $m=f$, $c=0.1f$	Measured processing rate
1	$f+2m$	$m+2c$	$f / (m+2c)CP$	132.3	136.3
2	$f+3m$	$m+2c$	$f / (m+2c)CP$	132.3	112.7
3	$2f+3m$	$m+3c$	$2f / (m+3c)CP$	244.2	243.0
4	$2f+4m$	$2m+2c$	$2f / (2m+2c)CP$	144.3	171.2
5	$3f+3m$	$m+f+3c$	$3f / (m+f+3c)CP$	207.0	207.8
6	$3f+4m$	$2m+3c$	$3f / (2m+3c)CP$	207.0	207.8
7	$3f+5m$	$2m+3c$	$3f / (2m+3c)CP$	207.0	207.7
8	$5f+4m$	$m+2f+3c$	$5f / (m+2f+3c)CP$	240.5	245.0
9	$7f+5m$	$m+3f+3c$	$7f / (m+3f+3c)CP$	258.4	242.3

Table 8. The number of actual operations, number of effective operations, estimated processing rate, estimated processing rate for $c = 0.1f$ (f is a floating point operation including fetching and storing the operands and c is a chaining operation), and measured processing rate for the basic operations on the ETA10-Q for vector length 256 and stride 1.

Loop No.	Actual operations	Effective operations	Estimated proc. rate (f_p)	f_p for $c=0.1f$	Measured proc. rate
1	f	$0.5f$	$f\ /\ (0.5f)CP$	105.3	90.9
2	f	$0.5f$	$f\ /\ (0.5f)CP$	105.3	93.5
3	$2f$	$0.5f+c$	$2f\ /\ (0.5f+c)CP$	175.4	171.3
4	$2f$	$0.5(2f)$	$2f\ /\ (f)CP$	105.3	93.5
5	$3f$	$0.5(2f)+c$	$3f\ /\ (f+c)CP$	143.5	131.8
6	$3f$	$0.5(2f)+c$	$3f\ /\ (f+c)CP$	157.9	135.2
7	$3f$	$0.5(3f)$	$3f\ /\ (1.5f)CP$	105.3	88.1
8	$5f$	$0.5(4f)+c$	$5f\ /\ (2f+c)CP$	125.3	107.0
9	$7f$	$0.5(5f)+2c$	$7f\ /\ (2.5f+2c)CP$	136.5	122.7

The Cray-2 has two floating point functional units per processor, and memory operations can be performed simultaneously with the floating point operations. Multiple functional units can help to reduce the number of effective operations only for complicated expressions, such as loops 8 and 9. Overlapping of CPU and memory operations resulted in a reduction in the number of effective operations for loops 4 through 9, see Table 6. No parallelism was found in loops 1 through 3. Another factor is the cost of these operations. A memory fetch of 64 elements on the Cray-2 takes 108 CPs to complete assuming there is no memory conflict, due to non-unit strides or other programs competing for memory. However, data transfer operations on the Cray-2 can overlap, and a second fetch operation can be initiated 72 CPs after the first. A floating point operation on the Cray-2 takes 79 CPs to complete. Nevertheless, the unit will be free 4 CPs later, and a second floating point operation can be issued. It was estimated that an average memory operation costs about 50% more than a floating point operation on a light load system. Based on this assumption, the estimated processing rates are within 10% of the

measured values.

The Cray Y-MP has all the four features that promote parallelism (listed at the beginning of the section). These features helped significantly in reducing the number of effective operations for all loops, see Table 7. Chaining was utilized at least twice in every loop. Also, multiple memory ports (four ports: one for I/O, two for fetching, and one for storing) helped in increasing memory bandwidth on the Cray Y-MP, and hence reducing the number of effective operations for every loop. A memory fetch of 64 elements on the Cray Y-MP takes 69 CPs to complete assuming there is no memory conflict. A floating point multiply operation takes 72 CPs while an add operation takes 71 CPs. Therefore, it is reasonable to assume that a memory operation costs as much as a floating point operation on the Cray Y-MP. Chaining costs a variable number of clock periods depending on the units involved in the process. It was assumed that chaining costs about 10% of a memory operation. Based on these assumptions, the estimated processing rates are within 15% of the measured values.

The ETA10-Q is a memory-to-memory vector computer, unlike the Crays which are register-to-register computers. A floating point operation is performed on the ETA10-Q by fetching two vectors from the processor memory, using two input data links, performing the operation, and storing the result back to the processor memory, using an output data link. In the vector unit of the CPU, there are two vector pipelines. Both pipelines perform identical operations, but only one type of operation at one time. For 64-bit operations, one pipe performs operations on the odd elements in the vector, the other performs on the even elements. This means that two results can be produced every clock period. That is why the

number of effective operations is multiplied by half in Table 8. Chaining on the ETA10-Q is possible only if two successive vector operations use different functional units and one of the operands is a scalar. This feature helped reducing the number of effective operations for loops 2, 5, 6, 8, and 9. It was assumed that chaining costs about 10% of a floating point operation. Based on this assumption, the estimated processing rates exceeded the measured rates by up to 20%.

The differences between the estimated and measured processing rates for the three machines can be attributed to several factors. Among these are the fluctuation of the timing results, the overlapping of memory operations between successive iterations of the outer loops, and the startup time of each outer loop. Also, there are other software factors such as operating systems: paging, caching, swapping, optimization by compilers, etc.

5. Comparison and discussion

The processing rate is one of the measures that can be used to compare the performance of these machines. Table 9 shows some interesting statistics for the performance results given in Tables 3 through 5. The basic operations achieved up to 39%, 78%, and 81% of the peak performance rates of the Cray-2, Cray Y-MP, and ETA10-Q, respectively. The lowest achieved rates represent 2.3%, 16%, and 2.4% of the peak performance rates of the Cray-2, Cray Y-MP, and ETA10-Q, respectively. The highest rates were achieved with stride 1 while the lowest were with stride 4.

Table 9. Performance comparison of the three machines.

I. Performance rates (Mflops):

Measure	Stride	Cray-2	Cray Y-MP	ETA10-Q
Highest achieved performance	1	191.0	247.8	171.3
Lowest achieved performance	1	38.3	102.5	46.8
Highest achieved performance	2	80.7	247.0	44.4
Lowest achieved performance	2	18.0	90.4	9.1
Highest achieved performance	4	48.6	216.6	25.5
Lowest achieved performance	4	11.1	52.2	5.0

II. Ratio of performance rates:

Measure	Stride	Y-MP/Cray-2	Y-MP/ETA10	Cray-2/ETA10
Highest perf. ratio	1	3.4	4.3	2.3
Lowest perf. ratio	1	1.1	1.2	0.5
Highest perf. ratio	2	5.7	12.2	2.8
Lowest perf. ratio	2	3.0	5.4	1.2
Highest perf. ratio	4	4.9	12.7	2.7
Lowest perf. ratio	4	4.5	7.6	1.6

The performance results, given in Tables 3 to 5 and summerized in Table 9, show that the Cray Y-MP outperformed the other two machines in every case. It outperformed the Cray-2 by up to 5.7 times and the ETA10-Q by up to 12.7 times. The Cray-2 outperformed the ETA10-Q by up to 2.8 times for non-unit stride operations. However, for stride 1, the Cray-2 outperformed the ETA10-Q by up to

2.3 times in some cases while the latter outperformed the former by up to 2.2 times in other cases. These differences can be attributed to the features that promote parallelism, discussed in the previous section, and the clock speed, which will be examined later. Beside these factors, there are architectural differences in handling non-unit stride operations by these machines.

Each Cray-2's CPU has one port to the main memory. Each memory quadrant has a data path to each of the four memory ports. Through its port, a CPU can access any given quadrant but only in the CPU's own phase time, that is, every forth clock period. Each port has four quadrant buffers, one for each quadrant, and a ten-slot backup buffer. A quadrant buffer can hold two memory references for its quadrant. When a quadrant buffer is filled, and another reference to that quadrant is made, the memory port begins a backup procedure for 10 CPs. This problem is called *quadrant conflict.*

Quadrant conflicts occur with even memory strides. For stride 2, addresses arrive at the quadrant buffers every 2 CPs but require 4 CPs to clear. After 4 CPs, both buffer slots are full and the fifth address causes a quadrant conflict. Backup occurs every fifth address for an effective transfer rate of 15 CPs for every 4 words. As shown in Table 3, stride 2 resulted in over 50% performance degradation on the Cray-2. Similarly, with stride 4, every address is in the same quadrant. Backup occurs every third address for an effective transfer rate of 13 CPs for every 2 words. Stride 4 resulted in over 70% performance degradation on the Cray-2 for the basic operations, see Table 3.

Memory strides of 2 and 4 do not cause a serious problem on the Cray Y-MP. This is because every CPU can access every memory section during every clock

period. The slight degradation in the performance of the Cray Y-MP for strides of 2 and 4 can be attributed to the attempt by multiple ports of the same CPU to simultaneously access the same memory section.

The ETA10-Q, like its Cyber predecessor, has a problem with non-unit memory strides. All elements of a vector must be read from memory even though only a small fraction may be operated upon. As shown in Table 5, stride 2 caused about 70% performance degradation on the ETA10-Q. Also, stride 4 caused about 85% performance degradation on the machine.

Another measure of performance is the number of clock periods required to execute a vector operation. This measure reduces the impact of technology on the performance of the machine. This measure favors the ETA10-Q since its clock speed is 3 times slower than the Cray Y-MP and 4.6 times slower than the Cray-2. However, the ETA10-Q is not the fastest model of the ETA10 series of supercomputers. (The ETA10-G, a super-cooled model introduced recently, has a 7 nsec clock period.) The ETA10-Q outperformed the Cray-2 in all cases by as much as 12 times using this measure. However, for stride 1, it outperformed the Cray Y-MP by a factor of up to 2.5 in most cases (21 out of 27) while the latter outperformed the former by up to 40% in other cases. For non-unit strides, the Cray Y-MP outperformed the ETA10-Q in all cases by as much as 4 times using this measure. This measure also favors the Cray Y-MP over the Cray-2, but the former has already outperformed the latter in the processing rate measure. This means that the Cray Y-MP would outperform the Cray-2 by a bigger factor using this measure; see Tables 3 and 4.

6. More tests: numerical algorithms

The use of basic operations for performance comparison can be criticized because they fail to measure the overhead associated with a complex algorithm and its embodiment in a specific program. In order to get a better feel for the performance of these machines, two numerical algorithms were also employed. The first algorithm is a four color cell relaxation scheme for the solution of the Cauchy-Riemann equations, see [5] for details. This scheme is equivalent to an SOR scheme. The second algorithm is an ADI scheme for the solution of the diffusion equation, see [6] for details. The ADI scheme consists of two sweeps to advance the solution one full step in time. Each sweep requires the solution of a set of tridiagonal systems.

Table 10 contains the processing rates (in Mflops) and the execution times (in million CPs) of the two algorithms on the Cray-2, Cray Y-MP, and ETA10-Q for three problem sizes: 64×64, 128×128, and 256×256 grid points. The relaxation algorithm requires 141, 327, and 771 iterations for convergence for the 64×64, 128×128, and 256×256 problems, respectively. Each iteration has 66 floating point operations per grid point. The ADI algorithm was run for 100 time steps. The two sets of the tridiagonal systems were solved by Gaussian elimination which has 15 floating point operations per grid point. For each case, the same code was run on the three machines. The inner loops of each code were fully vectorized on these machines. The results reported in Table 10 present the best case times for many runs.

Table 10. Processing rate (in Mflops) and execution time (in million clock periods) of the numerical algorithms on the three machines.

Domain size (points)	Cray-2		Cray Y-MP		ETA10-Q	
	Proc. rate (Mflops)	Exec. time (MCPs)	Proc. rate (Mflops)	Exec. time (MCPs)	Proc. rate (Mflops)	Exec. time (MCPs)
Cauchy-Riemann Eqs.						
64×64	109.6	82.2	177.6	33.0	22.3	87.1
128×128	112.9	751.7	190.1	290.6	26.1	702.2
256×256	115.2	7002.7	190.6	2755.7	28.1	6203.4
Diffusion Eq.						
64×64	85.6	33.9	130.8	14.4	26.3	23.8
128×128	88.9	132.7	135.2	56.8	31.3	81.4
256×256	95.2	500.0	136.5	226.8	34.4	298.4

The two algorithms achieved 18% to 24%, 41% to 60%, and 11% to 16% of the peak performance rates of the Cray-2, Cray Y-MP, and ETA10-Q, respectively. The algorithm for solving the Cauchy-Riemann equations has many memory operations with stride 2, which causes a significant performance degradation on the Cray-2 and ETA10-Q. The stride 2 problem, lack of chaining, and limited one path to memory are the reasons for achieving the given rates on the Cray-2. The results on the Cray Y-MP are quite reasonable for vectorized codes. The results on the ETA10-Q were less than expected. These results can be attributed to the lack of chaining for these codes and stride of 2 memory operations. Also, the immaturity of the software under the Unix environment may have an impact on the performance of the ETA10-Q.

The Cray Y-MP outperformed the other two machines for the three domain sizes of both algorithms. It outperformed the Cray-2 by over 2 times and the ETA10-Q by 4 to 8 times. Also, the Cray-2 outperformed the ETA10-Q by about 3 to 5 times. The clock period measure shows that the Cray Y-MP outperformed the ETA10-Q for all cases even though this measure favors the latter over the

former. Using the clock speed measure, the ETA10-Q outperformed the Cray-2 by up to 68% in five cases while the former outperformed the latter by 6% in one case; see Table 10.

7. Conclusions

A set of basic operations was developed to analyze the performance of three vector supercomputers: Cray-2, Cray Y-MP, and ETA10-Q. The results for three vector lengths, ranging from 64 to 256, and using memory strides of 1, 2, and 4 were analyzed and compared. These results showed that Cray Y-MP outperformed the Cray-2 and ETA10-Q in every one of the test cases. For stride one operations, the Cray Y-MP outperformed the Cray-2 by up to 3.4 times and the ETA10-Q by up to 4.3 times. Unlike the Cray-2 and ETA10-Q, even-numbered strides do not cause a major performance degradation on the Cray Y-MP. A memory stride of 2 resulted in an average performance degradation of 7% on the Cray Y-MP, compared to 55% on the Cray-2, and 73% on the ETA10-Q. A memory stride of 4 resulted in an average performance degradation of 36% on the Cray Y-MP, compared to 72% on the Cray-2, and 85% on the ETA10-Q. A performance model was developed to determine the impact of certain architectural features on the performance of these machines. The model showed that multiple ports to memory and chaining are the main reasons for achieving high performance on the Cray Y-MP. Two numerical algorithms were also implemented on the three machines to get better feel for the performance of these machines. These algorithms also showed the superiority of the Cray Y-MP over the other two machines. For three problem sizes of both algorithms, the Cray Y-MP outperformed the Cray-2 by over 2 times and the ETA10-Q by 4 to 8 times.

The Cray-2 has two important features which were not fully utilized in this study. These are the size of the main memory and the use of the local memory. Applications which need more than 3 Mwords of memory do not achieve high per-

formance on the ETA10-Q. This is because these applications require the use of the main memory with a data transfer rate of 1/6 of the local memory. Similarly, applications which need more than 30 Mwords of memory require the use of the SSD memory on the Cray Y-MP with some performance degradation. The local memory on the Cray-2, which has an access time of few clock periods compared to 45 CPs for the main memory, can also be used to store temporary variables. Although the local memory was partially used in the two numerical algorithms, it can be utilized more efficiently for large codes.

The results reported here were for basic operations and two simple numerical algorithms. More experiments may be needed to provide a better understanding of the different factors influence the performance of these machines. Full applications may be used to provide a better picture of the performance of different machines. The performance model can also be used in more complicated expressions and routines to identify the bottlenecks and limiting factors of different architectures.

This study showed that certain architectural features, such as multiple memory links and chaining, may have as much impact on the performance of a machine as its clock speed. We saw that for unit stride dyadic operations (loops 1 and 2), for example, the Cray Y-MP outperformed the Cray-2 by more than two times even though the Cray-2 has a faster clock than the Cray Y-MP. It is becoming quite hard and expensive to reduce the clock period much farther. Adapting these features and developing new ones for next generation supercomputers may be a more effective and less expensive approach.

Acknowledgements

I would like to thank Rebecca Popelka of CRI for her assistance in using the simulator on Cray machines.

References

[1] Anderson, R., Grimes, R., and Simon, H., "Performance comparison of the Cray X-MP/24 with SDD and the Cray-2," J. Supercomputing, 1, 1988, pp. 409-419.

[2] Cray-2 computer system functional description, HR-2000, Cray Research, Inc., 1986.

[3] Cray Y-MP computer systems functional description manual, HR-4001, Cray Research, Inc., 1988.

[4] ETA10 system reference manual, PUB-1005, ETA Systems, Inc., 1987.

[5] Fatoohi, R. A. and Grosch, C. E., "Implementation of a four color cell relaxation scheme on the MPP, Flex/32 and Cray/2," Proc. 1987 Int. Conf. Parallel Processing, pp. 424-426.

[6] Fatoohi, R. A. and Grosch, C. E., "Implementation of an ADI method on parallel computers," J. Scientific Computing, Vol. 2, No. 2, 1987, pp. 175-193.

[7] Hockney, R. and Jesshope, C., "Parallel computers 2: architectures, programming and algorithms," Adam Hilger, Bristol, England, 1988.

[8] Kampe, F. and Nguyen, T., "Performance comparison of the Cray-2 and Cray X-MP on a class of seismic data processing algorithms," J. Parallel Computing, 7, 1988, pp. 41-53.

[9] Luback, O., Moore, J., and Mendez, R., "A benchmark comparison of three supercomputers: Fujitsu VP-200, Hitachi S810/20, and Cray X-MP/2," Computer, Vol. 18, No. 12, 1985, pp. 10-24.